Perl による簡易版辞書ツールの作成法

著者	北村 達也
雑誌名	甲南大学総合研究所叢書
巻	111
ページ	1-21
発行年	2012-03-01
URL	http://doi.org/10.14990/00003048

Perlによる簡易版辞書ツールの作成法

北　村　達　也

1．はじめに

Reading Tutorの「辞書ツール」は日本語の文章を形態素解析し，それによって得られる形態素に訳語や例文をリンクし，Webブラウザ上で表示するシステムである[1][2]．また，e-chutaは利用者が形態素解析の結果や訳語などを修正できる機能を追加したものである[3]．これらのシステムはPerlというプログラミング言語[4][5]で開発されている．

Perlは文字列処理のプログラムを比較的容易に作ることができるプログラミング言語である．従って，日本語教育関係者がPerlを使いこなせるようになれば，教材作成や研究の効率を大幅に向上させることができる．そこで，本稿では，Perlを用いて簡易版辞書ツールを作成する過程を解説することによって，日本語教育分野でプログラミングを活用するヒントを提供する．なお，本稿ではWindows 7 を対象にして解説を行うが，Windows VistaやWindows XPでもほぼ同様に実行することができる．

本稿では，ファイルの中の文を形態素に分解し，各形態素を電子辞書（Jim BreenのEDICT）で検索し，得られた英訳を表示する簡易版辞書ツールを作成する．

以降では，まず環境設定について図を交えて解説し，次にプログラムの作成法を述べる．

2．Perlのインストール

以下，ソフトのインストールにはWindowsの管理者権限が必要である．

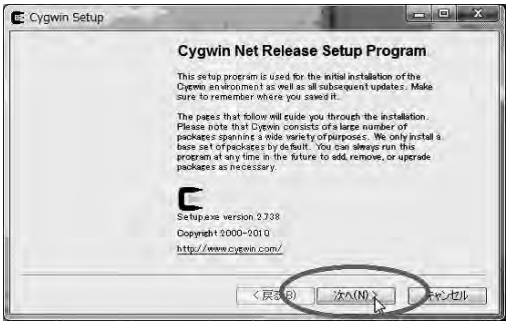

図1：Cygwinのセットアップ（1）

まずはWindowsにPerlをインストールする必要がある．WindowsでPerlを実行できるようにするにはいくつかの方法があるが，本稿ではCygwinを用いる．CygwinはWindows上でUNIXのコマンドなどを実行する環境を提供するシステムで，Perlもこの中で利用することができる．マウスの利用を中心としたWindowsの操作と異なり，Cygwinはキーボードからコマンドを入力することによって操作する．始めのうちはこの操作法の違いに拒絶反応を起こす人もいるかもしれない．

まず，CygwinのWebページ(1)にアクセスし，そのWebページの左側にあるInstall Cygwinの部分をクリックする．すると，Cygwin Installationというタイトルのページが現れるので，そのページからsetup.exeというファイルをダウンロードする．このファイルはCygwinをインストールするためのソフトウェア（インストーラー）である．

ダウンロードしたsetup.exeをダブルクリックし，Cygwinのインストールを開始する．まず，図1のダイアログが表示されるので，「次へ（N）」をクリックする．その後，いくつかのダイアログが順次表示されるので，これらも全て「次へ（N）」をクリックする．図2のダイアログでは，http://ftp.jaist.ac.jp/を選択して「次へ（N）」をクリックする．図3のダイアログでは「OK」をクリックし，次へ進める．なお，Windows XPでは本稿と図の表示順が多少異なる．

（1）http://www.cygwin.com/

引き続き，Perlのインストールを指定する．図 3 のダイアログにて「OK」をクリックすると，図 4 のインストールするソフトウェアを指定するダイアログが現れる．ここでまずInterpretersの文字をクリックし，次に図 5 のようにPerlのNewの列が数字になるまでクリックする．

その後，図 6 ， 7 ， 8 のダイアログで「次へ（N)」をクリックすればインストールが実行される．最後に図 9 のダイアログで「完了」をクリックする．

図 2 ：Cygwinのセットアップ（ 2 ）

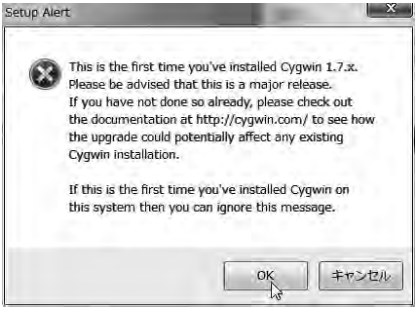

図 3 ：Cygwinのセットアップ（ 3 ）

図 4 ：Cygwinのセットアップ（ 4 ）

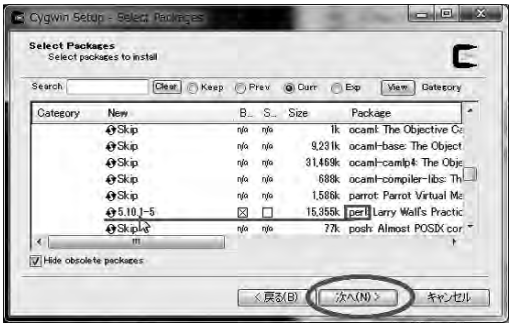

図 5 ：Cygwinのセットアップ（ 5 ）

図 6 ：Cygwinのセットアップ（ 6 ）

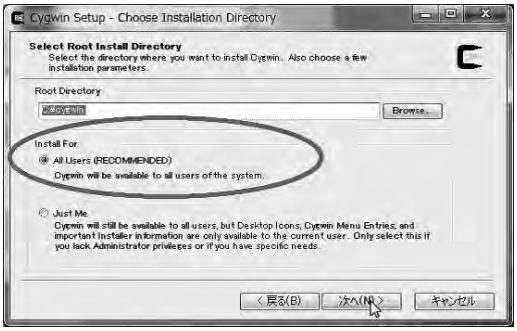

図 7 ：Cygwinのセットアップ（ 7 ）

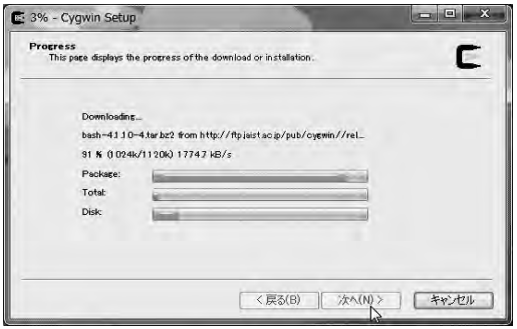

図 8 ：Cygwinのセットアップ（ 8 ）

図 9 ：Cygwinのセットアップ（ 9 ）

3．MeCab のインストール

MeCabは形態素解析ソフトウェアである．このソフトウェアは，文章を形態素に分割するとともに，形態素の基本形やよみや品詞などの情報を出力することができる．本稿で作成する簡易版辞書ツールでは，MeCabにより形態素の基本形を得て，電子辞書で検索する．

MeCabはhttp://sourceforge.net/projects/mecab/files/mecab-win32/0.98/からダウンロードできる．ダウンロードするファイル（インストーラー）はmecab-0.98.exeである．なお，このインストーラーには形態素解析のための辞書データも含まれる．これをダウンロードし，指示に従ってインストールする．辞書の文字コード指定のダイアログ（図10）ではUTF-8を指定する．

図10：MeCabのインストールにおけるコード指定

4．環境変数の設定

まず，スタートメニューからコンピューターを右クリックし，図11のようにプロパティを選択する．次に図12のウィンドウの右下の「設定の変更」をクリックする．Windows XPでは，スタートメニューからマイコンピューターを右クリックし，プロパティを選択するとシステムのプロパティの画面（図12）が表示される．

システムのプロパティの画面で「詳細設定」タブを開き，画面右下の「環境

変数」をクリックする（図13）．ユーザーの環境変数の「新規」をクリックし（図14），新しいユーザー変数の画面で変数名にPATH，変数値にC：¥Program Files¥MeCab¥binと入力し，OKをクリックする（図15）．

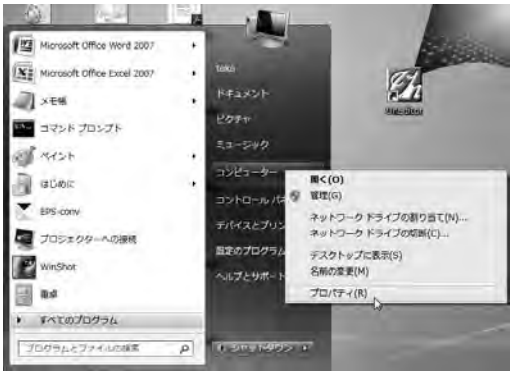

図11：環境変数の設定（1）

また，ユーザーの環境変数の変数名にHOMEがない場合は，ユーザーの環境変数の「新規」をクリックし新しいユーザー変数の画面で変数名にHOME，変数値にC：¥Users¥kitamura¥documentsのように入力し（ただし，kitamuraの部分は各自のユーザー名で置き換える），OKをクリックする[(2)]（図16）．Windows XPではC：¥Documents and Settings¥kitamura¥My Documentsのように入力する．ユーザーの環境変数にすでにHOMEはあるものの値がC：¥Users¥kitamura¥documents（Windows XPではC：¥Documents and Settings¥kitamura¥My Documents）のようになっていない場合は，「編集」をクリックし，修正する．

5．エディターのインストール

MS Wordなどのワープロソフトは図や表の挿入や箇条書きなど様々な機能

（2）ユーザー名はWindows 7ではスタートメニューの右上，Windows XPではスタートメニューの一番上に表示されている．

を有しているが，単にプログラムを書く目的には不必要な機能も多く，動作も遅い．一方で，Windowsにあらかじめインストールされているメモ帳には文字コードや改行コードを変換する機能がないなど機能不足である．そこで，本稿ではエディターソフト（以下単に「エディター」と書く）の利用を勧める．

エディターは編集機能に特化したソフトウェアで，プログラミングに便利な機能を持っているものも多い．本稿では一例としてUnEditor(3)を紹介しておく．

図12：環境変数の設定（２）

（３）http://www.yokkasoft.net/

図13：環境変数の設定（３）

図14：環境変数の設定（４）

図15：環境変数PATHにMeCabのフォルダを指定

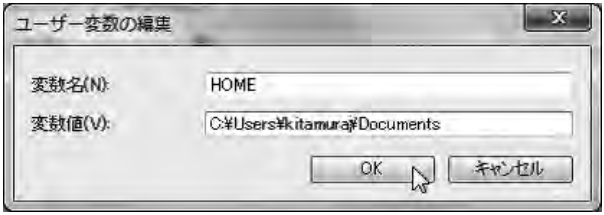

図16：HOMEを確認

6．MeCab の実行

6．1　フォルダの作成

まず作業用のフォルダを作成する．本稿では一例としてマイドキュメントの下にjisho_toolというフォルダを作ることにする．Windowsの通常の新規フォルダと同じようにマウスを使って作ればよい．

6．2　テキストファイルの準備

UnEditorを使って形態素解析したい文が書かれたファイルを作成し，jisho_toolに保存する．文は「本日は晴天なり。」などなんでもよい．ファイル名には日本語文字やスペースを用いず，input_text.txtなど英数字だけで構成されるようにする．ファイルを保存するとき,「文字コード」の欄は「Unicode（UTF-8BOM無）」を選択する（図17）．

図17：UnEditorでのファイル保存

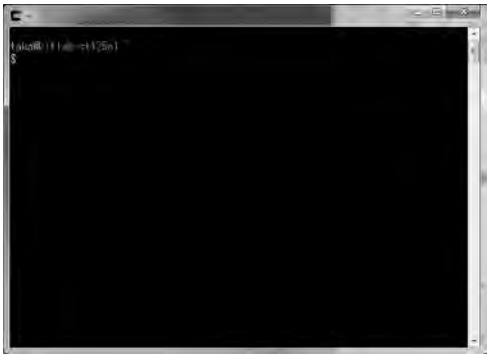

図18：Cygwinのウィンドウ

6．3　Cygwin　の起動と作業フォルダの移動

デスクトップにあるCygwinのショートカットやスタートメニューからCygwinを起動する．Cygwinが起動すると，図18のような黒いウィンドウが表示され，「$」が表示される．これを「プロンプト」という．

MS Wordやメモ帳など一般のソフトウェアと異なり，Cygwinでは作業対象

```
$ pwd
/cygdrive/c/Users/kitamura/documents
```

図19：pwdの実行（Windows 7 ）

```
$ pwd
/cygdrive/c/Documents and Settings/kitamura/My Documents
```

図20：pwdの実行（Windows XP）

```
$ cd jisho_tool
$ pwd
/cygdrive/c/Users/kitamura/documents/jisho_tool
```

図21：作業フォルダをjisho_toolに変更（Windows 7 ）

```
$ cd jisho_tool
$ pwd
/cygdrive/c/Documents and Settings/kitamura/My Documents/jisho_tool
```

図22：作業フォルダをjisho_toolに変更（Windows XP）

のファイルが存在するフォルダに移動して作業を行う必要がある[(4)]．4章の設定を間違いなく済ませていれば，Cygwinが起動した状況での作業フォルダは「マイドキュメント」になっている．そのことを確認するには，プロンプトにpwdを入力し，Enterキーを押せばよい．pwdはprint working directoryの略で，現在の作業フォルダを出力するコマンドである．その結果，図19（Windows

（4）ファイルのパスを指定すれば作業対象のファイルが存在するフォルダに移動しなくても作業できる．

７）や図20（Windows XP）のように（ただし，kitamuraの部分は各自のユーザ名で置き換える）表示されていれば問題ない．もし/cygdrive/c/home/kitamuraのように出力された場合には，４章の環境変数HOMEに関する設定が正しくできていないと考えられるので，やり直す．

Cygwinが起動した状況で作業フォルダが図19や図20のようになっていたら，作業フォルダを先ほど作成したフォルダに変更する．作業フォルダを変更するには，プロンプトにcd jisho_toolと入力し，エンターキーを押す．cdとjisho_toolの間にはスペースを１つ入力する．cdはchange directoryの略で，作業フォルダを移動するためのコマンドである．作業フォルダが変更できたことを確認するため，上述のpwdを再度実行する．その結果，図21（Windows７）や図22（Windows XP）のように（ただし，kitamuraの部分は各自のユーザ名で置き換える）表示されていれば問題ない．

６．４　MeCabの実行

MeCabの実行に先立ち，MeCabに関する設定が正しくできているか確認する．プロンプトにwhich mecabと入力しエンターキーを押す．その結果，/cygdrive/c/Program Files/MeCab/bin/mecabと表示されれば問題ない（図23）．MeCabのインストールが成功したにもかかわらず，which：no mecab in（以下略）のように出力される場合には，４章の環境変数PATHに関する設定が誤っているので修正する．

次に，図24のように入力し，エンターキーを押すことでMeCabが実行される．このコマンドは，input_text.txtという入力ファイルをmecabで処理して，その結果をresult.txtという出力ファイルに保存するものである．出力ファイル名は適宜変更してよいが，英数字で構成するようにする．結果を確認するには出力ファイルをエディターで開けばよい．ファイルには図25のように各行に１つの形態素が記載されており，各行には，形態素，よみ，基本形，品詞などの情報が記載されている．なお，EOSは解析結果の最後に出力されるものである．

```
$ which mecab
/cygdrive/c/Program Files/MeCab/bin/mecab
```

図23：MeCabが正しく設定されているか確認

```
$ mecab input_text.txt > result.txt
```

図24：MeCabの実行

```
本日　　名詞，副詞可能，＊，＊，＊，＊，本日，ホンジツ，ホンジツ
は　　助詞，係助詞，＊，＊，＊，＊，は，ハ，ワ
晴天　　名詞，一般，＊，＊，＊，＊，晴天，セイテン，セイテン
なり　　助動詞，＊，＊，＊，文語・ナリ，基本形，なり，ナリ，ナリ
。　　記号，句点，＊，＊，＊，＊，。，。，。
EOS
```

図25：MeCabの出力

7．EDICT のダウンロードと文字コードの変換

EDICTのデータはhttp://ftp.monash.edu.au/pub/nihongo/edict.zipからダウンロードする．ダウンロードした圧縮ファイルedict.zipを展開するとedictというフォルダが生成される．その中のedictというファイルをエディターで開き，１行目を削除する．そして，文字コード「UTF-8（BOM無）」，改行コード「LF ONLY」とし，edict.txtというファイル名でjisho_toolフォルダに保存する．なお，圧縮ファイルを展開するソフトがインストールされていない場合には，Lhaplusなどをインストールする．

8．プログラム作成

8．1　第 1 段階のプログラム

まずはプログラムの中でMeCabを実行し結果を表示するプログラムを作ってみる．エディターで図26の通りに入力し，mytool.plというファイル名でjisho_toolフォルダに保存する．このプログラムはCygwinプロンプトから図27のように入力して実行する．プログラムが正しく入力されていれば，result.txtにMeCabの処理結果が出力される．

このプログラムの 1 行目では，入力ファイルをMeCabで処理した結果を@resultという「配列」に代入している．配列は複数のデータを 1 つずつ保存できる，たんすのような入れ物をイメージすればよい．その引き出しには番号がふってあり，番号を指定することによってそれに対応する引き出しの中身を知ることができるというイメージである．@resultには図25のMeCabの出力が保存されていることになる．なお，入力ファイル（input_text.txt）は，$ARGV［0］で指定されている．

2 行目から 4 行目までは，その@resultの各行に対する処理になる．各行を一旦$rという「変数」に代入し，その$rの中身をprintコマンドで画面に出力している．複数のデータが保存される配列と異なり，変数には 1 つのデータしか保存できない．foreachは繰り返し処理をするためのコマンドで，@result

```
@result = ‘mecab $ARGV[0]‘ ;
foreach $r (@result) {
        print $r ;
}
```

図26：第 1 段階のプログラム

```
$ perl mytool.pl input_text.txt > result.txt
```

図27：mytool. plの実行方法

の最後にたどりつくまで ｛ ｝ の中の処理を繰り返す．

8．2 第2段階のプログラム

第2段階のプログラムでは，MeCabの結果から形態素だけを取り出して出力する．図26のプログラムを図28のように変更し，図27のようにして実行する．なお，3行目の／\s／の\はキーボードの￥キーで入力する．\が￥と表示されていても問題ない．

このプログラムの3行目では，変数＄rの中身をスペースを境にして分割し，配列@items 1 に保存している．splitは，指定した文字を境に変数の中身を分割するコマンドで，splitの中の／\s／はスペースを境にして分割することを指定している．そして，4行目では，分割して配列に代入した要素の先頭の要素を画面に出力している．Perlでは配列の要素を指定する数字は0から始まることになっているので，＄items 1［0］が第1番目の要素を意味する．なお，配列の各要素には1つのデータしか含まれないので，@items 1 の第1番目の要素には＄をつけて＄items 1［0］と表す．

```
@result = ‘mecab $ARGV[0]‘ ;
foreach $r (@result) {
        @items1 = split(/\s/, $r) ;
        print $items1[0] ;
}
```

図28：第2段階のプログラム

8．3 第3段階のプログラム

第2段階のプログラムを実行すると，文が終わったことを示すEOSも表示されてしまっていた．第3段階のプログラムは，このEOSが表示されないようにしたものである．

4行目のifから始まる行は「if文」と呼ばれるもので，条件によって処理を変えたいときに利用する．ここでは，@items 1 の第1番目の要素がEOSでな

ければ，それを画面に表示するという処理を行っている．ifに続くカッコの中身は，この「@items 1 の 1 つめの要素がEOSでなければ」という条件を指定している．neはnot equalの略である．

```
@result = ‘mecab $ARGV[0]‘ ;
foreach $r (@result) {
        @items1 = split(/\s/, $r) ;
        if ($items1[0] ne ”EOS”) {
                print $items1[0] ;
        }
}
```

図29：第 3 段階のプログラム

8．4　第 4 段階のプログラム

1 章で述べたように，簡易版辞書ツールでは各形態素を辞書で検索しその英訳を形態素に付与する．辞書は基本形で検索する必要があるので，MeCabの出力から基本形を取り出す必要がある．図30の第 4 段階のプログラムでは，MeCabの出力から基本形を取り出して，形態素の後にカッコで囲んで表示させている．

```
@result = ‘mecab $ARGV[0]‘ ;
foreach $r (@result) {
        @items1 = split(/\s/, $r) ;
        if ($items1[0] ne ”EOS”) {
                print $items1[0] ;
                @items2 = split(/,/, $items1[1]) ;
                print ”($items2[6])” ;
        }
}
```

図30：第 4 段階のプログラム

8．5　第5段階のプログラム

この段階ではEDICTのデータを読み込み，「ハッシュ」に保存する．ハッシュは配列と同様，複数のデータを1つずつ保存できるたんすのような入れ物である．しかし，ハッシュの場合は引き出しに番号がふってあるのではなく，文字列を書いたラベルが貼ってあり，その文字列を指定することによってそれに対応する引き出しの中身を知ることができるというイメージである．辞書データは，単語に対応する英訳を保存しておく必要があるので，このハッシュを用いるのが都合がよい．

プログラムの1行目では，文字コード変換済みのEDICTのファイルedict.txtを読み込むために，ファイルを開く（オープンする）という操作をしている．

```
open(EDICT, ”edict.txt”) ;
foreach $e (<EDICT>) {
    chomp($e) ;
    @items3 = split(/] /, $e) ;
    @items4 = split(/\s/, $items3[0]) ;
    $dictionary{$items4[0]} = $items3[1] ;
}
close(EDICT) ;

@result = ‘mecab $ARGV[0] ‘ ;
foreach $r (@result) {
    @items1 = split(/\s/, $r) ;
    if ($items1[0] ne ”EOS”) {
        print $items1[0] ;
        @items2 = split(/,/, $items1[1]) ;
        print ”($items2[6])” ;
    }
}
```

図31：第5段階のプログラム

これをしないとプログラムからファイルの中身にアクセスすることができない．また，ファイルの中身へのアクセスが終了したらファイルを閉じる（クローズする）必要がある（8行目）．

2行目から7行目は，edict.txtの各行について処理を行っている．まず，3行目で行末についている改行コードを取り除いている．そして，4行目では「/」を境に分割し，@items 3に代入している．5行目では4行目で分割した1つ目の要素（$items 3［0］）をさらにスペースを境に分割し，@items 4に代入している．6行目ではハッシュdictionaryにデータを記録し，見出し語（$items 4［0］）と訳語（$items 3［1］）を対応づけている．これにより，見出し語を指定すると訳語が得られるようになる．

この段階で追加したプログラムにprintの行がないので，このプログラムを実行しても出力は第4段階と同じである．

8．6　プログラム完成

第5段階で作った辞書データのハッシュdictionaryを使って，入力文章の形態素を辞書で検索する（図32）．上述のように，見出し語を指定すると訳語が得られるので，16行目では形態素の基本形（$items 2［6］）を指定して訳語（$eng）を得ている．そして，それが空でない場合（17行目）にカッコではさんで出力している．

入力として「本日は晴天なり。」を与えた場合の処理結果（result.txtの中身）を図33に示す．「本日」と「晴天」にEDICTが付与されていることがわかる．

```
open(EDICT, ”edict.txt”) ;
foreach $e (<EDICT>) {
    chomp($e) ;
    @items3 = split(/] /, $e) ;
    @items4 = split(/\s/, $items3[0]) ;
    $dictionary{$items4[0]} = $items3[1] ;
}
close(EDICT) ;

@result = ‘mecab $ARGV[0]‘ ;
foreach $r (@result) {
    @items1 = split(/\s/, $r) ;
    if ($items1[0] ne ”EOS”) {
        print $items1[0] ;
        @items2 = split(/,/, $items1[1]) ;
        $eng = $dictionary{$items2[6]} ;
        if ($eng ne ””) {
            print ”($eng)” ;
        }
    }
}
```

図32：完成版のプログラム

```
本日 (/(n−adv,n−t)today/(P)/) は晴天 (/(n)fine weather/(P)/) なり。
```

図33：「本日は晴天なり。」を入力した場合の出力結果

9．おわりに

以上，プログラミング言語Perlで簡易版辞書ツールを作成する手順を解説した．わずか20行程度でこのような処理が可能である．紙数の制約から細かい説明を省いたところはあるが，これを入り口にPerlの本で勉強していただければ幸いである．教科書として，結城による「新版Perlプログラミングレッスン入門編」[4]と高橋による「ゼロからわかるPerl言語超入門」[5]を紹介しておく．なお，完成版のプログラムなどの情報を当研究室のWebページ(5)に掲載しておくので活用していただきたい．

謝辞

本研究は甲南大学総合研究所，平成23年度科学研究費補助金基盤研究（B）（21320095）の支援を受けています．なお，２章の一部の画像は甲南大学自然科学研究科山本晴日さん，内田聖也さんによるCygwinのインストール方法に関するWebページの画像を利用させていただきました．また，本稿の執筆には甲南大学知能情報学部 三木翔太さんと同文学部 中島知子さんの協力を得ました．

参考文献

［１］北村達也，川村よし子，内山潤，寺朱美，奥村学，学習履歴管理機能を持つ日本語読解支援システムの開発とその評価，日本教育工学会誌，23，127-133（1999）．
［２］川村よし子，北村達也，保原麗，EDR電子化辞書を活用した日本語教育用辞書ツールの開発，日本教育工学会誌，24(Suppl.)，7 -12(2000)．
［３］北村達也，荊木亜里沙，森川結花，永須実香，川村よし子，前田ジョイス，斉木美紀，金善子，Web上の読解教材作成を支援するツールの開発及び活用法，甲南大学情報教育研究センター紀要，10(2011)．
［４］結城浩，新版Perlプログラミングレッスン入門編，ソフトバンククリエイティブ(2006)．
［５］高橋順子，ゼロからわかるPerl言語超入門，技術評論社（2011）．

（５）http://basil.is.konan-u.ac.jp/